Useful Programs For Your TIMEX/SINCLAIR Computer
ZX81; T/S 1000; T/S 1500

★ MUSIC LIBRARY ★

Explaining the Program, Its Use, And Its Modification.

Kendric C. Smith

CONTENTS

Chapter 1. Introduction

This program allows you to store **three lines** of information (32 **characters** each) on 100 (or more) **items** of music (sheet music, records, tapes, etc.) **without reloading** the program. There are **routines** for **adding, deleting** or **correcting** information, for **alphabetizing** the information according to the **first line** (title?), and for **searching** for **any word** on all **three** lines (OMNISEARCH). The information from your searches or sorts can be **printed** to the screen or to a printer. Ways of modifying this program to increase its storage capacity and usefulness are included.

If you have suggestions for improving this program or the manual, please let me know.

Kendric C. Smith
927 Mears Court
Stanford, CA 94305

Chapter 2. Using The MUSIC LIBRARY

Before you use this program, you should give careful thought as to how you want to format your library. The examples on the example tape show you one type of format, but you are not limited to this format. The present format gives the title of the music first, then the artist, then line 3 is for you to use as you wish. If there are 70 → no entries for line 3, just key ENTER and this line will remain blank, 75ch → and the computer will proceed to the next command. This program alphabetizes on the first line of your entry (title?). If you would rather have it alphabetize by artist, enter the names, last name first, on line 1. So, organize your music on paper first so that you will be happy with it when you enter it into the computer.

This program is written for 16k of RAM (16,384 bytes), and has so many "user friendly" features that information on only 100 pieces of music (records, tapes, etc.) can be stored without reloading or modifying the program. However, as will be mentioned in Chapter 5, there are several routines in the program that can be deleted or modified in order to save memory so that more music can be added. Alternatively, you could save **100 items on one tape**, and **100 items** on **another tape**. If you have 32k of RAM, you could save about 280 items on one tape (since all of the extra 16k of RAM would be used for memory), but you would first have to change the statements on lines 101-115, 130, and 135 of the program.

An example of the print-out of this program is shown below. The **number** in the **first** line is the **"N" value**, that is, the counter that the computer uses to keep track of all the entries **(1-100)**. The N value does not stay with the same listing if you alphabetize the list. Putting the **N value** on your music would allow you to **store** your music **in sequence**, and thus make it easier fo find in your cabinet.

32

60

3 LINES
32 CHAR EACH LINE

```
6
THE COMPLETE LIONEL HAMPTON
1937-1941.  (6 RECORDS)
BLUEBIRD AXM6-5536
```

TAPE #
DATE
DISCRIPTION +
TITLES

60 → To load the program, key in LOAD "ML" ENTER. Since this is an auto-start program, the first menu will appear on the screen after the program has loaded. You are offered five choices:

1. ENTER MUSIC 1ST TIME?
2. ADD MUSIC?
3. CORRECT OR DELETE?
4. PRINT?
5. SAVE MUSIC LIBRARY?

ENTER MUSIC 1ST TIME. By keying 1 and ENTER, you are prompted as to what to enter into the MUSIC LIBRARY:

ENTER LINE 1 (maximum of 32 spaces)
ENTER LINE 2 (maximum of 32 spaces)
ENTER LINE 3 (maximum of 32 spaces)

If you have **no entries** to make on a line (for example, on line 3), just key **ENTER** and this line will **remain blank**, and the computer will move onto the next command.

After entering all the information on one item, the computer prints the information on the screen, and asks you if the entries are **"ALL CORRECT? (Y/N)"**. If the answer is Yes (enter Y), the computer then asks if you have **"ANY MORE ENTRIES? (Y/N)"**. If the answer to this question is Yes, the computer continues with the prompts for you to add all the information for the next entry; if the **answer is No** (enter N), then the computer **returns** to the **first menu**.

If the answer to the first question is No, that is, the entries are not correct, then the **"WHAT IS INCORRECT?"** menu comes on the screen below the listing of the entries. You then enter the appropriate number from the menu.

To **correct** an entry, **you must reenter the complete string** (that is, B$, C$, or D$), that is, the whole line must be entered rather than just correcting one letter, as you would on a word processor.

After entering your list of music, **never use RUN to start the program**, since this **will erase the memory**. If you are in the program listing, you can regain the first menu by keying **GOTO 1 ENTER**.

After entering your first list of music you can add line 100 to the program; **key 100 GOTO 70.** This added line will **prevent you** from **erasing all** of your entries by **accidently entering 1** on the first menu.

SAVE your MUSIC LIBRARY **before** you **try** out any of the **search** and **sort** routines. Then, if the computer crashes, you can just reload instead of having to reenter all the music.

ADD MUSIC. This routine functions the same as the first routine. It just **continues to add items to the end** of the list that you entered originally.

CORRECT OR DELETE. To execute this routine, you must **enter the N value** for the item to be corrected or deleted. Then you just follow the **"WHAT IS INCORRECT?"** menu, as above.

PRINT? Again you are offered a menu:

1. PERFORM A SEARCH
2. PRINT WHOLE LIST
3. ALPHABETIZE WHOLE LIST
4. RETURN TO FIRST MENU

PERFORM A SEARCH. The first question that you will see on the screen is "PRINT TO SCREEN OR PRINTER?(S/P)". If you want the latter, then you must **turn on** your **printer** before **keying P** and ENTER.

In this routine you **search** for **LEN Q$**, that is, for the length of the string ($) (that is, word or part of word) that you enter. Thus, LOUIS ARMSTRONG would be found by entering L, LO, LOU, LOUI, LOUIS, etc. Of course all the other words that contain these letters would also be found. To make an exact search you would need to enter the complete name. This is an **OMNISEARCH** routine, and will print out any music item that contains, **on any line**, the word (or part of word) **that you type in for** Q$. For this reason the search is somewhat slow. It takes 22 seconds to search through 10 listings. Be patient.

Four items of music fill the screen; then a **prompt** appears at the bottom of the screen to tell you to **enter CONT** in order TO **CONTINUE LIST**. In the print-to-printer routine these steps are bypassed, and the complete list is printed without stopping. If you are in the print-to-screen mode and want to copy to the printer what is on the screen, just key COPY ENTER.

After all the items that came from your search have been printed (S or P), a prompt appears on the screen to tell you to enter CONT in order to regain the "PRINT A LIST?" menu.

Line **1015** sets a counter, Q, to zero. If nothing is found during the search, Q will remain equal to zero, and then **line 1072** will be executed: **NOT FOUND**.

__PRINT WHOLE LIST__. As the name implies, this routine prints the whole list of items. The information concerning print-to-screen or print-to-printer, etc., are the same as described above for the SEARCH routine.

__ALPHABETIZE__. This routine sorts (alphabetizes) by the **first line** of music (titles?) (B$). Note that when the list is alphabetized, the **N** values **change** for each item. The **N** values are needed in order to CORRECT or to DELETE an item. After the music has been sorted in alphabetical order, the computer switches to the "PRINT WHOLE LIST" routine.

__RETURN TO FIRST MENU__. This routine needs no explanation.

Chapter 3. Explaining the Program

The listing of the program for the MUSIC LIBRARY is given in Chapter 4. Each of the subroutines in the program are explained below. Ways to modify this program are discussed in Chapter 5.

Lines **10-50**: First Menu. This subroutine lists all the major options available in the program, and asks you to enter the number that corresponds to the subroutine that you want to execute.

Lines **55-75**: This routine tells the computer to go to the subroutine that will execute the function that you chose from the first menu. If you enter 1 to 5, the correct program will be executed. If you enter numbers 6 to Ø, the computer executes lines **70** and **75**, that is, it lists the **first menu again**. It does this because the conditions in line 65 are not satisfied. If you enter the letter Q, the same result occurs. **Line 70** isn't needed at this stage, but it is needed later in the program to get back to the first menu.

If you enter a letter that has not been used in the program, the computer prints the report code: **2/55** (**undefined variable** at **line** 55). If you key ENTER you now have access to the program listing. If you want to go back to the first menu, just key **GOTO** 1 ENTER. **Never use RUN (or CLEAR)** after entering your list of music, or you will erase everything in the memory.

If you enter a letter that has been used as a variable in the program, then the screen will go blank and a few white horizontal lines will appear; the computer is trying to execute a routine but doesn't really know what to do, and will eventually crash. If you key BREAK you will stop the computer and get the **report code: D/9990.** Key **GOTO 1 ENTER** to regain the first menu.

Lines **105-175**: ENTER MUSIC FIRST TIME. The **DIM** (DIMension) statements in this routine set aside space in the computer's memory for the various strings ($) that you wish to store. With this "user friendly" program and 16k of memory, there is only room for information on 100 items. More items could be accommodated, for example, if some of the space set aside in the strings is reduced, and if some of the user friendly routines are eliminated. You can change this program to fit your needs (see **Chapter** 5).

B$ is generally for the title; C$ is generally for the artist; and D$ is for whatever you want to enter (for example, a second artist, or notes on the music).

Lines **135, 160, 165, 170, 175** are a subroutine to tell you that the **memory is full**, in case you try to enter item number 101.

Line **140** sends you to a subroutine for **entering information** into the MUSIC LIBRARY (see lines **630-700**).

Line **145** **tells** the computer to list on the screen the information that you have just entered (see lines **720-745**). The next routine (line **750**) allows you to correct this entry if necessary (see lines **750-805**).

Line 200: ADD MUSIC. Line 200 tells the computer to **GOTO 130**, which skips the DIM statements. (If the command went to the DIM statements, this would erase the music stored in the memory; see above.) This routine brings up the next N value just as if you were entering items for the first time, and **adds new items** at the **end of the list.**

Lines **300-380**: CORRECT OR DELETE. You must **enter** the **N** value of the listing to be changed or deleted. Line **315** executes GOSUB 360, which prints on the screen the listing that goes with value of N that you entered. When line **380** RETURNs the computer to line **320**, it executes GOSUB **900**, which is the "WHAT IS INCORRECT?" menu. This menu will be discussed under line 900.

Lines **400-450**: This is the "PRINT A LIST?" menu, and is self explanatory. Each of these routines (1 through 4) will be discussed below (lines 1000, 2000, 3000, 4000).

Lines **455-475**: See lines **55-75** for an explanation.

Lines **500-505**: SAVE MUSIC LIBRARY. Line 500 clears the screen, and line 505 goes to the end of the program (line 9990) to SAVE "ML". Line 9995 tells the computer to GOTO 1. This is a **self-starting** routine. When you load this program (LOAD "ML"), the first menu comes up on the screen, and not the program listing. See Chapter 5 for how to "break" into the program listing.

Lines **630-700**: This subroutine prompts you to enter the various **items** of information programmed for this MUSIC LIBRARY. After each prompt it waits for your input.

Lines **720-745**: After going through the subroutine for entering information (see lines 630-700), this routine makes the last N value one number larger, and then "prints" on the screen what you have just typed in (see line **130** for how D relates to N).

Lines **750-805**: This routine then asks if what you typed in the routine given in lines 630-700 is "ALL CORRECT?". If the answer is Yes (**line 760**), then the computer asks you if you have any more entries (Y/N) (**line** 785). If the answer is Yes (line 800), then the computer RETURNs to line 150; if the answer is No (that is, it is not Yes), the computer goes to **line 70** (line 805), that is, returns to the first menu.

If the answer to the first question (ALL CORRECT?) is No, then the computer executes line 770 (**GOTO 900**), which is the "WHAT IS INCORRECT?" menu, discussed below.

Line 900: WHAT IS INCORRECT? Line 900 is the complete menu. Note how commas and spaces have been used so that the menu prints out in the same style as the first menu, but this routine uses less memory.

Lines **905-975**: This subroutine tells you what to enter, depending upon what you want to correct. Remember, that to correct a line you must **reenter the whole line.**

Lines **970** and **975** clear the screen and print out on the screen the listing that you just corrected, and then again asks you if t is correct.

If you **enter 4**, then the computer will execute CLS **and GOSUB 5000**. Line 5000 begins the deletion routine.

If nothing is wrong, **enter 5** and the computer will return to the first menu.

Lines 1000-1004: This routine asks you whether you want to print-to-screen or print-to-printer. Depending upon what you enter (S or P), it sends you to the correct print routine.

Lines 1008-1070: This search routine is written to search for LEN Q$. Thus, the computer will print out any listing that contains a word (or part of a word) that you type in for Q$. It reads through each string one letter at a time to see if the next set of letters equals Q$. Since you can't read past space 32, the FOR NEXT loops (lines 1025, 1037, 1047) go only to (32-LEN Q$).

Line 1072: If the counter, Q, in line 1015 has not been incremented, that is, if nothing has been found in the search, then Q remains equal to zero, and line 1072 is executed.

Lines 1080-1090: After all the items that came from your search have appeared on the screen (or the printer), this routine comes on the screen to tell you to enter CONT in order to regain the "PRINT A LIST?" menu.

Lines 2000-2020: See lines 1000-1004.

Lines 2050-2070: PRINT WHOLE LIST. Line 2050 sets a counter, Q, to zero. This counter is used later in the print-to-screen routine (lines 9300-9330). This routine then executes the print routine that you selected (S or P).

Lines 3000-3125: ALPHABETIZE. This is the classical "bubble sort" routine. In this case we are sorting (alphabetizing) the first line (generally titles), that is, B$(N). The rest of the program, from line 3075 on, is just to make sure that all the information on a given title "follows" the title in the memory and the printing, after it has been rank ordered by alphabetizing.

Line 4000: RETURN TO FIRST MENU. This routine needs no explanation.

Lines 5000-5080: DELETE. You have to enter an N value in order to delete the listing with that N value. From that N value to the last entry of the directory, D, the computer moves the listing for N + 1 down to N, thus erasing the listing for N, etc. After moving down the whole list, the value for D is reduced by one number.

Lines 9200-9220: This is the print-to-printer routine.

Lines 9300-9317: This is the print-to-screen routine, and except for substituting PRINT for LPRINT, is the same as the print-to-printer routine above.

Lines 9320-9330: This routine prints only 4 complete listings on the screen at a time. Otherwise, the computer would just print as many lines as it could to fill the screen, and part of the fifth listing would be missing. One determines what to divide "Q" by (that is, how many listings will appear each time), by being aware of the number of lines per listing (including blank lines between the listings), and the number of available lines on the screen. This routine is not needed for the print-to-printer routine, so line 9220 is RETURN rather than GOTO 9320.

Lines 9335-9345: This is just a routine to remind you to enter CONT in order TO CONTINUE LIST.

Lines 9990-9995: SAVE. See explanation under lines 500-505.

Line 9999: Memory Left. If you key GOTO 9999 ENTER, this routine prints on the screen how many bytes of useable memory are left out of 16k (actually, 16,384 bytes) of RAM. This routine is very helpful when you want to keep track of the remaining useable memory while you are modifying a program. The present program has 2086 bytes of RAM left to use.

REM
LINE 100 GOTO 70

16K STORES 100 RECORDS
32K " 280
64K " 460

Chapter 4. The Program Listing

VIDIO TAPES

"VT"

```
  10 PRINT AT 1,5;"~~MUSIC LIBRARY~~
""VT"""
  15 PRINT AT 2,14;"BY"
  20 PRINT AT 3,7;"KENDRIC C. SM
ITH"
  25 PRINT AT 6,9;"ENTER NUMBER"
  30 PRINT AT 9,1;"1  ENTER MUSI
C 1ST TIME?"
  35 PRINT AT 12,1;"2  ADD MUSIC
?"
  40 PRINT AT 15,1;"3  CORRECT O
R DELETE?"
  45 PRINT AT 18,1;"4  PRINT?"
  50 PRINT AT 21,1;"5  SAVE MUSI
C LIBRARY?"
  55 INPUT Q
  60 CLS
  65 IF Q<=5 THEN GOSUB Q*100
  70 CLS
  75 GOTO 10 180 64
 105 DIM B$(100,32)
 110 DIM C$(100,32)
 115 DIM D$(100,32)
 125 LET D=0
 130 FOR N=D+1 TO 100
 135 IF N=101 THEN GOTO 160
 140 GOSUB 600
 145 GOSUB 720
 150 NEXT N
 160 PRINT AT 11,10;"MEMORY FULL
"
 165 PRINT AT 21,0;"ENTER ""CONT
"" TO REGAIN MENU"
 170 STOP
 175 GOTO 70
 200 GOTO 130
 300 PRINT AT 11,0;"ENTER VALUE
OF ""N"" OF LISTING TO BE CHANGE
D OR DELETED"
 305 INPUT N
 310 CLS
 315 GOSUB 360
 320 GOSUB 900
 325 GOTO 300
 360 PRINT N
 362 PRINT B$(N)
 365 PRINT C$(N)
 370 PRINT D$(N)
 375 PRINT
 380 RETURN
 400 PRINT AT 1,8;"PRINT A LIST?
"
 405 PRINT AT 3,8;"ENTER NUMBER"
 410 PRINT AT 5,1;"1  PERFORM A
SEARCH"
 440 PRINT AT 7,1;"2  PRINT WHOL
E LIST"
 445 PRINT AT 9,1;"3  ALPHABETIZ
E WHOLE LIST"
 450 PRINT AT 11,1;"4  RETURN TO
FIRST MENU"
 455 INPUT Q
 460 CLS
 465 IF Q<=4 THEN GOSUB Q*1000
 470 CLS
 475 GOTO 400
 500 CLS
 505 GOTO 9990
```

180

200/64 64 192

(9..)

505

```
 630 PRINT AT 11,0;"ENTER LINE 1
"
 635 INPUT B$(N)
 637 CLS
 650 PRINT AT 11,0;"ENTER LINE 2
"
 655 INPUT C$(N)
 657 CLS
 660 PRINT AT 11,0;"ENTER LINE 3
"
 665 INPUT D$(N)
 667 CLS
 700 RETURN
 720 CLS
 725 LET D=D+1
 730 PRINT N
 732 PRINT B$(N)
 735 PRINT C$(N)
 740 PRINT D$(N)
 745 PRINT
 750 PRINT AT 7,0;"ALL CORRECT?
(Y/N)"
 755 INPUT Q$
 760 IF Q$="Y" THEN GOTO 780
 770 GOTO 900
 780 CLS
 785 PRINT AT 6,0;"ANY MORE ENTR
IES? (Y/N)"
 790 INPUT Q$
 795 CLS
 800 IF Q$="Y" THEN RETURN
 805 GOTO 70
 900 PRINT AT 7,0;"WHAT IS INCOR
RECT?","1 LINE 1",,"2 LINE 2",,"
3 LINE 3",,"4 EVERYTHING; DELETE
 LISTING","5 NOTHING"
 905 INPUT Q
 920 IF Q=1 THEN INPUT B$(N)
 925 IF Q=2 THEN INPUT C$(N)
 930 IF Q=3 THEN INPUT D$(N)
 955 IF Q=4 THEN CLS
 960 IF Q=4 THEN GOTO 5000
 965 IF Q=5 THEN GOTO 70
 970 CLS
 975 GOTO 730
1000 PRINT AT 11,0;"PRINT TO SCR
EEN OR PRINTER?(S/P)"
1001 INPUT Q$
1002 CLS
1003 IF Q$="S" THEN LET Z=9300
1004 IF Q$="P" THEN LET Z=9200
1008 PRINT AT 11,0;"ENTER WORD(S
) TO BE SEARCHED"
1009 INPUT Q$
1010 CLS
1015 LET Q=0
1020 FOR N=1 TO D
1025 FOR I=1 TO (32-LEN Q$)
1030 IF Q$<>B$(N,I TO ((I-1)+LEN
 Q$)) THEN NEXT I
1035 IF Q$=B$(N,I TO ((I-1)+LEN
Q$)) THEN GOSUB Z
1037 FOR I=1 TO (32-LEN Q$)
1040 IF Q$<>C$(N,I TO ((I-1)+LEN
 Q$)) THEN NEXT I
1045 IF Q$=C$(N,I TO ((I-1)+LEN
Q$)) THEN GOSUB Z
1047 FOR I=1 TO (32-LEN Q$)
```

1035

1046

↑

SEE PAGE 12
FOR NEW LINES

[illegible]

```
1050 IF Q$<>D$(N,I TO ((I-1)+LEN
 Q$)) THEN NEXT I
1055 IF Q$=D$(N,I TO ((I-1)+LEN
Q$)) THEN GOSUB Z
1070 NEXT N
1072 IF Q=0 THEN PRINT AT 11,6;"
NOT FOUND"
1075 GOTO 1080
1080 PRINT AT 21,0;"ENTER ""CONT
"" TO REGAIN MENU"
1085 STOP
1090 RETURN
2000 PRINT AT 11,0;"PRINT TO SCR
EEN OR PRINTER?(S/P)"
2005 INPUT Q$
2010 CLS
2015 IF Q$="S" THEN LET Z=9300
2020 IF Q$="P" THEN LET Z=9200
2050 LET Q=0
2055 FOR N=1 TO D
2060 GOSUB Z
2065 NEXT N
2070 GOTO 1080
3000 FOR P=1 TO (D-1)
3005 LET N=P
3010 FOR C=P+1 TO D
3015 IF B$(C)>=B$(N) THEN GOTO 3
025
3020 LET N=C
3025 NEXT C
3030 LET S$=B$(P)
3035 LET B$(P)=B$(N)
3040 LET B$(N)=S$
3075 LET S$=C$(P)
3080 LET C$(P)=C$(N)
3085 LET C$(N)=S$
3090 LET S$=D$(P)
3095 LET D$(P)=D$(N)
3100 LET D$(N)=S$
3120 NEXT P
3125 GOTO 2000
4000 GOTO 70
5000 FOR L=N TO D-1
5020 LET B$(L)=B$(L+1)
5030 LET C$(L)=C$(L+1)
5040 LET D$(L)=D$(L+1)
5060 NEXT L
5070 LET D=D-1
5080 GOTO 70
9200 LPRINT N
9203 LPRINT B$(N)
9206 LPRINT C$(N)
9209 LPRINT D$(N)
9217 LPRINT
9220 RETURN
9300 PRINT N
9303 PRINT B$(N)
9306 PRINT C$(N)
9309 PRINT D$(N)
9317 PRINT
9320 LET Q=Q+1
9325 IF INT (Q/4)=Q/4 THEN GOTO
9335
9330 RETURN
9335 PRINT AT 21,0;"ENTER ""CONT
"" TO CONTINUE LIST"
9340 STOP
9345 RETURN
9990 SAVE ~~"M~~ "VT"
9995 GOTO 1
9999 PRINT AT 11,5;"MEMORY LEFT:
 ";16384-(PEEK 16404+256*PEEK 16
405-16283);" BYTES"
```

Take out space

C=P

VIDEOTAPES

65535

Chapter 5. Modifying the Program

A. Saving Memory.

There are two reasons for taking the time to save memory, and it does take time. One reason is to save bytes in the program itself so that you will have more memory for data storage and manipulation. This MUSIC LIBRARY program is a good example of the need to save memory in the program so that it can be used to store extra items.

The second reason for making the program as short as possible is to save time in the LOAD and SAVE modes. A full 16k RAM takes over 6 minutes to LOAD or SAVE. Thus, a program that you want to use frequently that is written to save bytes would save you a lot of LOAD time.

Anytime that you write a program and have a limited amount of useable memory, or you want to cut down on LOAD time, compromises will have to be made as to how "user friendly" or "goof-proof" you can afford to make the program.

If you wish to add additional items to this MUSIC LIBRARY after you have saved memory by some of the tricks listed below, or you have more than 16k of RAM, just change the number 100 in lines 105-115 and line 130 to some larger number "x", and then change the number 101 in line 135 to "x + 1". Going from 16k of RAM to 32k of RAM will more than double your useable memory for data storage, because the second 16k of RAM will only be needed for data storage.

(1) One place to save memory in this program is in the DIM statements (lines 105-115). For example, maybe you don't need 32 spaces for C$ and D$, and you could save memory here. Presently, each entry of music uses 32 x 3 = 96 bytes.

(2) Actually, 2086 bytes of useable memory are still available for this program. You could immediately increase the number of entries from 100 to 120 (100 + 2086/96 = 121.7). See above for directions for changing the DIM statements. Alternatively, you may want to use these extra bytes for a "fast load" program, which uses up RAM (500 to 1200 bytes, depending upon the brand).

(3) You can also save memory by deleting some of the routines used in this program. For example, you could save enough bytes for 4 more items if you delete the alphabetizing routine: delete lines 445, and lines 3000-3125.

(4) The word "enter" is used 11 times in this program. By substituting the word "type" for "enter", 11 bytes would be saved; by substituting "key" for "enter", 22 bytes would be saved. They all add up!

B. Improving the Program.

(1) If you want to get fancy and have ENTER NUMBER flash in the first menu, that is, cycle between the graphics display and the regular display mode, then add the following lines:

```
21  SLOW
22  FOR I = 1 TO 10

52  PRINT AT 6,9; "ENTER NUMBER"
53  NEXT I.

57  FAST
```

You regulate how long it flashes by the number you enter after "TO" in line 22. Of course, this routine uses up 80 bytes.

(2) If you want to print a list without the N value, then change line 9200 to LPRINT (that is, delete "N").

(3) If you want to alphabetize on the artist's name rather than the title, then just enter the artist's name, last name first, on line 1 when you first enter your MUSIC LIBRARY. Then your title, etc. would go on lines 2 and 3.

(4) Use the routine on line 9999 (GOTO 9999) to tell you how many bytes left in your program, when you make modifications.

C. Some Tips:

If you want to get to the program list, but are stuck on an INPUT routine, do the following: for L in the lower left hand corner of the screen, enter STOP ENTER ENTER; for "L" in the lower left corner, move the L to the left of the quotation marks, then proceed as above for L. If you are in a CONT mode, just key ENTER.

To Run Faster

```
1036  IF Q$ = B$ (N, I To ((I-1) + LEN Q$))
                    Then NexT N


1046  IF Q$ = C$ (same as abouE

1056           Samr For D$
```